

"Express Mail" mailing label number EV 331533820 US

Date of Deposit: March 30, 2004

ATTORNEY DOCKET NO.: 15619US02

**ELECTRONIC DEVICE NETWORK SUPPORTING ENCIPHERING AND  
DECIPHERING AND UPDATE GENERATION IN ELECTRONIC DEVICES**

**CROSS-REFERENCE TO OTHER APPLICATIONS**

[0001] The present application claims the benefit of priority of U.S. Provisional Patent Application having serial number 60/459,091, filed on March 31, 2003, and hereby incorporates herein by reference the complete subject matter thereof, in its entirety.

[0002] The present application also hereby incorporates herein by reference the complete subject matter of PCT Application having publication number WO 02/41147 A1, and having application number PCT/US01/44034, filed on November 19, 2001, in its entirety.

[0003] The present application also hereby incorporates herein by reference the complete subject matter of U.S. Provisional Patent Application having serial number 60/249,606, filed November 17, 2000, in its entirety.

**FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT**

[0004] [Not Applicable]

**[MICROFICHE/COPYRIGHT REFERENCE]**

[0005] [Not Applicable]

**BACKGROUND OF THE INVENTION**

[0006] Electronic devices (i.e., mobile electronic devices having software/firmware), for example, mobile cellular phones, personal digital assistants

(PDA's), pagers, MP3 players, digital cameras, etc. often contain firmware and/or application software that are either provided by the manufacturers of the electronic devices, telecommunication carriers, or third parties. These firmware and application software often contain bugs. New versions (updates) of the firmware and software are periodically made available to fix the bugs, introduce new features, delete features, etc.

[0007] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of skill in the art, through comparison of such systems with some aspects of the present invention as set forth in the remainder of the present application with reference to the drawings appended hereto.

## SUMMARY OF THE INVENTION

- [0008] Aspects of the present invention may be found in an electronic device network for updating at least one of firmware and software in a plurality of electronic devices using at least one electronic device update. At least one of the firmware and software in the plurality of electronic devices and the at least one update may be encrypted. The network may comprise at least one update generator adapted to generate updates. The at least one update generator may comprise an encrypting and decrypting engine. The network may also comprise at least one update store storing a plurality of electronic device updates, and at least one update delivery server adapted to dispense the plurality of electronic device updates.
- [0009] In an embodiment according to the present invention, the at least one update delivery server may comprise secure sockets layer support providing authentication and data encryption/decryption.
- [0010] In an embodiment according to the present invention, each of the plurality of electronic devices may be adapted to retrieve secure encrypted updates from the at least one update delivery server to update the at least one of firmware and software resident in the plurality of electronic devices. At least a portion of the at least one of firmware and software resident in the electronic devices may be encrypted.
- [0011] In an embodiment according to the present invention, each of the plurality of electronic devices may comprise one of encrypting and decrypting components, and a client for downloading updates.
- [0012] In an embodiment according to the present invention, each of the plurality of electronic devices may comprise a security services component providing secure communication with the at least one update delivery server.
- [0013] In an embodiment according to the present invention, each of the plurality of electronic devices may comprise an encrypted section. The encrypted section may comprise at least one of an encrypted data section and an encrypted code section.

[0014] In an embodiment according to the present invention, each of the plurality of electronic devices may comprise at least one of a random access memory, a provisioned data section, an operating system, an update agent, and an update application loader. The provisioned data section may comprise an update agent provisioning information section and a number assignment module.

[0015] In an embodiment according to the present invention, the update agent may be adapted to employ at least one of encrypting and decrypting components to update at least one of firmware and software resident in the electronic devices. At least a portion of the at least one of firmware and software may be encrypted and stored in one of an encrypted data section and an encrypted code section.

[0016] In an embodiment according to the present invention, the update generator may be adapted to process an old memory image and a new memory image of the at least one of firmware and software in the electronic devices. At least a portion of the at least one of firmware and software may be encrypted.

[0017] In an embodiment according to the present invention, the update generator may be adapted to decipher one of encrypted data segments and encrypted code in both an old memory image and a new memory image to generate an update for updating at least one of firmware and software in the electronic devices.

[0018] In an embodiment according to the present invention, the update generator may be adapted to employ deciphering techniques to extract one of enciphered code and enciphered data segments, process the one of enciphered code and enciphered data segments to generate an update comprising difference information, and encipher the one of code and data segments, and the difference information in at least one update.

[0019] In an embodiment according to the present invention, the electronic devices may comprise a plurality of mobile electronic devices. The plurality of mobile electronic devices may comprise at least one of a mobile cellular phone handset, personal digital assistant, pager, a multimedia player, and a camera.

[0020] Aspects of the present invention may be found in a method of encrypting update information within a firmware image in electronic devices. The method may comprise creating encrypted updates for an electronic device using binary differencing information, and encrypting firmware images by applying at least one of stream symmetric enciphering and block symmetric enciphering.

[0021] In an embodiment according to the present invention, stream symmetric enciphering may be performed in a byte by byte manner. Update information may be processed using a key stream to produce an encrypted update.

[0022] In an embodiment according to the present invention, stream symmetric enciphering may further comprise an  $i^{\text{th}}$  byte of the key stream operating on a byte of the update information produce an  $i^{\text{th}}$  cipher encrypted byte.

[0023] In an embodiment according to the present invention, the  $i^{\text{th}}$  cipher encrypted byte may be decrypted by the  $i^{\text{th}}$  byte of the key stream operating on the  $i^{\text{th}}$  cipher encrypted byte to reproduce an original  $i^{\text{th}}$  byte of update information.

[0024] In an embodiment according to the present invention, block symmetric enciphering may be performed upon blocks of data. The blocks of data may comprise a predetermined number of bytes. A key block may be applied to an update information block to produce an encrypted block. Block symmetric enciphering may be performed by cipher block chaining.

[0025] In an embodiment according to the present invention, the predetermined number of bytes in the blocks of data may comprise 8-16 bytes.

[0026] In an embodiment according to the present invention, block symmetric enciphering may be enabled to accommodate variable block sizes. Block sizes may at least be one of expanded and padded. Padding may be one of added and removed to vary the block sizes during a ciphering process.

[0027] In an embodiment according to the present invention, an enciphering algorithm and an enciphering key may be stored in the electronic devices.

[0028] In an embodiment according to the present invention, the electronic devices may comprise a plurality of mobile electronic devices. The plurality of mobile electronic devices may comprise at least one of a mobile cellular phone handset, personal digital assistant, pager, multimedia player, and a camera.

[0029] Aspects of the present invention may be found in an electronic device employing one of encrypting and decrypting techniques to update firmware and software. The electronic device may comprise random access memory and non-volatile memory. The non-volatile memory may comprise an update agent, a first in first out (FIFO) memory device, a firmware, a software application, and an update. The electronic device may be adapted to be updated by performing the update upon at least one of the firmware and the software application selected for updating.

[0030] In an embodiment according to the present invention, the at least one of the firmware and the software application selected for updating in the electronic device may be at least partially encrypted.

[0031] In an embodiment according to the present invention, the electronic device may be adapted to retrieve secure encrypted updates from an update delivery server to update at least one of the firmware and the software application selected for updating resident in the electronic device.

[0032] In an embodiment according to the present invention, the electronic device may comprise at least one of encrypting and decrypting components and a client for facilitating downloading updates.

[0033] In an embodiment according to the present invention, the electronic device may comprise a security services component providing secure communication with an update delivery server.

[0034] In an embodiment according to the present invention, the electronic device may comprise an encrypted section. The encrypted section may comprise at least one of an encrypted data section and an encrypted code section.

[0035] In an embodiment according to the present invention, the electronic device may further comprise at least one of a provisioned data section, an operating system, an update agent, and an update application loader. The provisioned data section may comprise an update agent provisioning information section and a number assignment module.

[0036] In an embodiment according to the present invention, the update agent may be adapted to employ at least one of encrypting and decrypting components to update at least one of firmware and software application resident in the electronic device. At least a portion of the at least one of firmware and software application may be encrypted and stored in one of an encrypted data section and an encrypted code section.

[0037] In an embodiment according to the present invention, the electronic device may comprise a plurality of mobile electronic devices. The plurality of mobile electronic devices may comprise at least one of a mobile cellular phone handset, personal digital assistant, pager, multimedia player, and a camera.

[0038] Aspects of the present invention may be found in a method of building a firmware upgrade for use in an electronic device incorporating encryption. The method may comprise building a firmware image to be encrypted. The firmware image may comprise a plurality of components. The method may also comprise encrypting the components before assembling the components into an encrypted firmware image.

[0039] In an embodiment according to the present invention, the method may further comprise generating binary difference information between firmware versions undergoing an upgrade and using an un-encrypted firmware image to generate the binary difference information. As the upgrade is being applied to an encrypted firmware image, uncorrelated information may be decrypted.

[0040] In an embodiment according to the present invention, the method may further comprise creating a data update package. The data update package may be based upon un-encrypted binary images.

[0041] In an embodiment according to the present invention, the method may further comprising creating a data update package. The data update package may be based upon encrypted binary images.

[0042] In an embodiment according to the present invention, the method may further comprising at least one of managing encrypted information by performing a pre-check analysis, managing encrypted information by performing a check-recovery analysis, and managing encrypted information by performing a fault tolerant procedure.

[0043] In an embodiment according to the present invention, during at least one of the pre-check analysis and the check recovery analysis, a cyclic redundancy check of a firmware image block may be compared against an original image cyclic redundancy check stored in a data update package. When ciphered data is present, the pre-check analysis may be performed upon the block to be decrypted before the cyclic redundancy check is calculated.

[0044] In an embodiment according to the present invention, cyclic redundancy check values for ciphered data may be stored in the data update package.

[0045] In an embodiment according to the present invention, during the fault tolerant procedure a ciphering algorithm may be applied to facilitate recovery of data for the upgrade.

[0046] In an embodiment according to the present invention, the method may further comprise decrypting an original data block and copying the decrypted data block to random access memory, applying update information to the random access memory, the update information comprising at least one of an update code and an update data segment from a data update package, combining the decrypted data block and the update information forming an update, encrypting the update information forming an encrypted update, sending the encrypted update to a storage unit, overwriting the original data block with the encrypted update, and processing every data block to be updated.

[0047] In an embodiment according to the present invention, a fault tolerant upgrade may comprise maintaining each original data block intact until the original data

block is overwritten by an encrypted updated data block and maintaining a data update package intact throughout the fault tolerant upgrade.

[0048] In an embodiment according to the present invention, the electronic device may comprise a plurality of mobile electronic devices. The plurality of mobile electronic devices may comprise at least one of a mobile cellular phone handset, personal digital assistant, pager, multimedia player, and a camera.

[0049] These and various other advantages and features of novelty which characterize the invention are pointed out with particularity in the claims annexed hereto and that form a part hereof. However, for a better understanding of the invention, its advantages, and the objects obtained by its use, reference should be made to the drawings which form a further part hereof, and to accompanying descriptive matter, in which there are illustrated and described specific examples of an apparatus in accordance with the invention.

## **BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS**

- [0050] **Figure 1** is a block diagram illustrating an electronic device employing enciphering and deciphering techniques to update firmware and software in electronic devices in accordance with an embodiment of the present invention;
- [0051] **Figure 2** is a block diagram illustrating an electronic device network employing enciphering and deciphering techniques to update firmware and software in electronic devices in accordance with an embodiment of the present invention;
- [0052] **Figure 3** is a block diagram illustrating a firmware build process incorporating ciphering according to an embodiment of the present invention;
- [0053] **Figure 4** is a block diagram illustrating a firmware update generator updating firmware images according to an embodiment of the present invention;
- [0054] **Figure 5** is a block diagram illustrating a firmware update generator updating firmware images using ciphered firmware images according to an embodiment of the present invention;
- [0055] **Figure 6** is a block diagram illustrating a modified update agent fault tolerant update process managing ciphered firmware images according to an embodiment of the present invention; and
- [0056] **Figure 7** is a flow diagram illustrating a method of updating firmware objects according to an embodiment of the present invention.

## **DETAILED DESCRIPTION OF THE INVENTION**

[0057] Aspects of the present invention may be found in a method of efficiently generating updates in electronic devices wherein at least a portion of content of the electronic device image is encrypted. Another aspect of the present invention may be found in a method of updating code and/or data segments in the electronic device when the code and/or data segments are encrypted. An aspect of the present invention may be found in determining what types of encrypted code and/or data segments may be updateable in the electronic device.

[0058] In an embodiment according to the present invention, it may be determined whether an update agent may be capable of updating all types of encrypted data and/or code segments resident in the electronic device. An aspect of the present invention may be found in an electronic device adapted to update encrypted firmware/software.

[0059] Electronic devices may be adapted to access servers to retrieve updates for updating at least one of firmware and software. The updates may be encrypted and/or the firmware/software being updated may be encrypted. An electronic device may be, for example, a mobile electronic device having software/firmware, such as, mobile cellular phone handsets, personal digital assistants (PDA's), pagers, multimedia players, cameras, etc.

[0060] An update may comprise firmware and software updates that modify or change the version of a particular firmware or software installed in an electronic device. For example, the updates may upgrade firmware/software to a newer version, repair a bug in the firmware/software, etc. An update may also add new services to the electronic device or delete services, as desired by a service provider, device manufacturer, or an end-user. The updates may be encrypted and/or the firmware/software being updated may be encrypted. An update and/or update package may comprise a set of executable program instructions for converting a first firmware/software version to a second firmware/software version.

[0061] **Figure 1** is a block diagram illustrating an electronic device 105 employing enciphering and deciphering (encrypting and decrypting) techniques to update firmware and software in electronic devices in accordance with an embodiment of the present invention. In **Figure 1**, the electronic device 105 may comprise a mobile handset 107, for example. The mobile handset 107 may comprise a random access memory (RAM) 125 and non-volatile memory 111.

[0062] The non-volatile memory 111 may comprise a plurality of components. For example, the non-volatile memory 111 may comprise an update agent 127, a first in first out (FIFO) memory device 113, a firmware 117, an operating system 119, an application (for example, a software application) 121, and an update 115. The electronic device 105 may be adapted to be updated by performing the update 115 upon the components (firmware/software) selected for update.

[0063] **Figure 2** is a block diagram illustrating an electronic device network 205 employing enciphering and deciphering (encrypting and decrypting) techniques to update firmware and/or software in electronic devices, for example, mobile handset 207, in accordance with an embodiment of the present invention. **Figure 2** illustrates an electronic device network 205 that may employ in electronic devices enciphering and deciphering techniques to update firmware and/or software that are partially or entirely enciphered. The updates may also be encrypted.

[0064] The electronic device network 205 may comprise an update generator 255 adapted to generate updates that may be employed to update firmware/software in electronic devices, for example, mobile handset 207. In an embodiment according to the present invention, the update generator 255 may comprise a ciphering/enciphering and deciphering (encrypting and decrypting) engine 257. The electronic device network 205 may also comprise an update store 253 storing a plurality of electronic device updates.

[0065] The electronic device network 205 may also comprise a delivery server 245 adapted to dispense the plurality of electronic device updates. The delivery server 245 may also comprise secure sockets layer (SSL) support 249 providing authentication and data encryption/decryption in an embodiment according to the present invention.

The electronic device network 205 may also comprise a plurality of electronic devices, for example, mobile handset 207. The plurality of electronic devices may be adapted to retrieve secure encrypted updates from delivery server 245 to update firmware/software resident in the electronic devices, wherein at least a portion of the firmware/software resident in the electronic devices may be enciphered/encrypted.

[0066] In an embodiment according to the present invention, the electronic devices, for example, mobile handset 207, may comprise ciphering/enciphering and deciphering (encrypting and decrypting) components 235 and a client 221 for facilitating downloading updates. The electronic devices, for example, mobile handset 207, may also comprise a security services component 223 providing secure communication with delivery server 245. The electronic devices, for example, mobile handset 207, may also comprise an encrypted section 237. The encrypted section 237 may also comprise a ciphered/enciphered data section 243 and a ciphered/enciphered code section 241.

[0067] The electronic devices, for example, mobile handset 207, may also comprise a random access memory (RAM) 225, a provisioned data section 229, an operating system (OS) 219, a firmware 217, an update agent 213, and an update application (UA) loader 227. The provisioned data section 229 may also comprise an update application/agent (UA) provisioning information section 231 and a number assignment module 233.

[0068] In an embodiment according to the present invention, the update agent 213 may be adapted to employ the ciphering/enciphering and deciphering (encrypting and decrypting) components 235 to update firmware 217 and/or software (not shown) resident in the electronic device. At least a portion of the firmware 217 and/or software (not shown) may be enciphered and stored in encrypted section 237, as illustrated in **Figure 2**.

[0069] An electronic device, for example, mobile handset 207, may comprise an update agent 213 adapted to decipher code and/or data segments and employ the deciphered code and/or data segments along with, and in conjunction with, contents of an

update to update firmware 217 and/or software components, operating system (OS) 219, etc., resident in the mobile handset 207.

[0070] In an embodiment according to the present invention, the update generator 255 may be adapted to process an existing (old) memory image and a new memory image of the firmware/software in the electronic devices, for example, mobile handset 207. At least a portion of the firmware/software may be enciphered employing ciphering techniques, such as block ciphering techniques.

[0071] The update generator 255 may decipher the enciphered content (data segments and/or code) in both of the existing (old) and the new memory images to generate an update that may be used to update the firmware/software image in the electronic devices, for example, mobile handset 207.

[0072] The update generator 255 may also employ deciphering techniques to extract enciphered code and/or data segments, process the code and/or data segments to generate an update comprising difference information, and encipher the code and/or data segments, difference information, in an update.

[0073] Aspects of the present invention may also be found in including ciphered/enciphered information within a firmware image. In an embodiment according to the present invention, ciphering/enciphering scrambles messages and causes similar firmware images to become apparently un-correlated. In an embodiment according to the present invention, binary differencing techniques may also be used in creating the firmware/software updates.

[0074] Ciphering/enciphering may comprise scrambling information to hide the actual content from un-intended viewers. Several methodologies of ciphering/enciphering exist. One method of ciphering/enciphering firmware images is by applying symmetric ciphering/enciphering. Symmetric ciphering/enciphering may use one or multiple keys for both encryption and decryption. Symmetric ciphering/enciphering may be performed in at least the following two embodiments: stream symmetric ciphering/enciphering and block symmetric ciphering/enciphering.

[0075] Stream symmetric ciphering/enciphering may operate on information in a byte by byte manner. A key stream may be combined with a message to produce a ciphered/enciphered message. Stream symmetric ciphering/enciphering may be mathematically represented, for example, as follows:

$$C[i] = KS[i] \otimes M[i] ; \text{Encryption}$$

$$M[i] = KS[i] \otimes C[i] ; \text{Decryption}$$

[0076] For example, to encrypt a message according to the stream symmetric ciphering/enciphering method of an embodiment of the present invention, the  $i^{\text{th}}$  byte of key stream KS may operate on a byte of message M to produce the  $i^{\text{th}}$  cipher C, for example. To decrypt a message, the  $i^{\text{th}}$  byte of key stream KS may operate on the  $i^{\text{th}}$  cipher C to reproduce a byte of the message M, for example. Stream symmetric ciphering techniques are known to have been employed in general cryptography, however, stream symmetric ciphering techniques are not known to have been previously used in techniques of updating firmware/software in mobile electronic devices.

[0077] An example of stream symmetric ciphering used in general cryptography, for example, is RC4 (Ron's Code or Rivest's Cipher level 4). RC4 is a stream ciphering technology designed by Ronald Rivest for RSA Security™. RC4 is a variable key-size stream ciphering technique having byte-oriented operations. The algorithm for RC4 is based on the use of a random permutation. Analysis shows that the period of the cipher using RC4 is likely to be greater than  $10^{100}$ . Eight to sixteen machine operations are required per output byte using RC4. The RC4 cipher and corresponding algorithm are considered secure.

[0078] Block symmetric ciphering/enciphering, in contrast to stream symmetric ciphering/enciphering, may break the information into blocks of data, instead of bytes. The blocks may comprise 8-16 bytes, for example. A key block may be applied to a message to produce ciphered/enciphered text. In an embodiment according to the present invention, block symmetric ciphering/enciphering may be performed by cipher block

chaining, for example. Block symmetric ciphering/enciphering may be mathematically represented, for example, as follows:

$$C[i] = E(K[i], M[i] \otimes C[i-1]) ; \text{Encryption}$$

$$M[i] = D(KS[i], C[i]) \otimes C[i-1] ; \text{Decryption}$$

[0079] For example, using block symmetric ciphering/enciphering, the message data may be expanded to accommodate variable block size requirements. For example, large messages may be impacted (e.g., expanded) toward the end of the message. Block symmetric ciphering techniques are known to have been employed in general cryptography, however, block symmetric ciphering techniques are not known to have been previously used in techniques of updating firmware/software in mobile electronic devices.

[0080] Examples of block symmetric ciphering techniques used in general cryptography are DES (data encryption standard), 3DES (triple-DES) and RC2 (Ron's Code or Rivest's Cipher level 2). RC2 is a variable key-size block cipher designed by Ronald Rivest for RSA Security™. (RC stands for "Ron's Code" or "Rivest's Cipher"). RC2 is faster than DES and is designed as a "drop-in" replacement for DES. RC2 can be made more secure or less secure than DES against exhaustive key search by using appropriate key sizes. RC2 has a block size of 64 bits and is about two to three times faster than DES. An additional string (40 to 88 bits long) called a salt can be used to thwart attackers who try to precompute a large look-up table of possible encryptions. The salt may be appended to the encryption key. The lengthened key is used to encrypt the message. The salt is then sent, unencrypted, with the message.

[0081] The critical information in a firmware image, or a portion thereof, may be ciphered/enciphered. A ciphered/enciphered firmware image may comprise code and data segments. Image ciphering/enciphering may be useful for protecting information. A symmetric ciphering/enciphering algorithm and key may be stored in the electronic device.

[0082] **Figure 3** is a block diagram 305 illustrating a firmware build process incorporating ciphering/enciphering according to an embodiment of the present invention. **Figure 3** illustrates a process of building a firmware image 350 to be ciphered/enciphered incorporating a plurality of components, for example, component n 330, a component comprising an ellipsis 331 indicating that the firmware image comprises a plurality of additional component not shown, component 1 332, component 2 333, and boot loader 334. The component(s) to be ciphered/enciphered may be encrypted before assembling into an encrypted firmware image 360. If ciphering/enciphering keys change firmware build processes, the ciphered/enciphered component(s) of firmware images may become apparently un-correlated. Producing binary difference information between two firmware versions may be influenced by encryption, wherein the difference information may be masked due to firmware/software upgrade.

[0083] In an embodiment according to the present invention, an un-encrypted firmware image, for example firmware image 350, may be used while generating the binary difference. As the update is applied to the firmware image, the uncorrelated information may be decrypted.

[0084] **Figure 4** is a block diagram 405 illustrating a firmware update generator 480 updating firmware images according to an embodiment of the present invention. In **Figure 4**, at least two methods of generating an update using the firmware update generator 480 are illustrated. In an embodiment according to the present invention, one method may comprise creating an data update package (DUP) 490 based upon un-encrypted binary images, for example, firmware image 1 460 and firmware image 2 470 illustrated in **Figure 4**, wherein additional information comprising ciphered/enciphered components may be used and included when ciphering/enciphering firmware images 1 and 2, 460 and 470, during update generation.

[0085] Symmetric ciphering/enciphering may have little or no impact on the final firmware image size. Stream symmetric ciphering/enciphering may produce ciphered/enciphered components having the same size as the un-encrypted components. Block symmetric ciphering, on the other hand, may introduce extra bytes at the end of the

firmware images. The extra bytes may represent padding information, wherein the end of the message may not be aligned with a cipher block boundary. The firmware update generator 480 may be unaffected, because an encryption/decryption algorithm may be adapted to add/remove the padding information.

[0086] **Figure 5** is a block diagram 505 illustrating a firmware update generator 580 updating firmware images, for example firmware image 1 560 and firmware image 2 570, using ciphered/enciphered (encrypted) firmware images, for example encrypted image 1 565 and encrypted image 2 575 according to an embodiment of the present invention. In order to perform ciphering/enciphering in an update agent in an electronic device, at least the following three issues may be addressed: handling ciphered/enciphering information during a pre-check phase; handling ciphered/enciphered information during a check-recovery phase; and handling ciphered/enciphered information using a fault tolerance procedure.

[0087] During the pre-check phase, the cyclic redundancy check (CRC) of a firmware image block may be compared against an original image CRC stored in the data update package (DUP) 590. When ciphered data is present, an approach to perform the pre-check may be for the block to be decrypted before the CRC is calculated.

[0088] In another embodiment according to the present invention, the CRC values for ciphered data may be stored in the DUP 590. However, in the firmware update generator, all images, ciphered/enciphered and un-encrypted, may be present during the data update package (DUP) generation stage.

[0089] The check-recovery phase may also perform a CRC comparison to determine the updated state applying the same processes used for the pre-check.

[0090] In the fault tolerance procedure, modifications may be performed, wherein a ciphering algorithm may be included to facilitate recovery of data for the update. Ciphering algorithm may comprise encryption, decryption, enciphering, deciphering, etc.

[0091] **Figure 6** is a block diagram 605 illustrating a modified update agent fault tolerant update process managing ciphered firmware images according to an embodiment

of the present invention. In **Figure 6**, a flash data block 630, for example, may be decrypted and copied to RAM as a decrypted data block 666. The update code and/or data segments may be added from the data update package (DUP) 650 into the RAM, wherein the decrypted data block may be updated with the update information. The updated decrypted data block 666 may be encrypted and sent to a working flash buffer 640 for backup storage. The encrypted update may be sent to and overwritten into the original flash data block 630 as an updated data block. The process illustrated above may be repeated for every data block or data byte, depending upon the ciphering/enciphering method, to be updated.

[0092] Aspects of the present invention may be found in a fault tolerant update process. In an embodiment according to the present invention, because the original data block remains intact until the block is overwritten by an updated and encrypted update block, the original data block may be used, if necessary, in a re-initiated update process, if an error occurs during update processing. Additionally, the data update package also remains intact throughout the update process. If a processing failure or error occurs during the update process, the failure is recoverable and the process may be re-initiated without loss of the original data block or loss of the data update package.

[0093] **Figure 7** is a flow diagram 705 illustrating a method of updating firmware objects according to an embodiment of the present invention. In **Figure 7**, A current object to be updated may be copied to RAM from a backup FIFO 710. The next object to be updated may also be copied to the backup FIFO 720. The blocks in RAM may be updated using the objects in the back up FIFO 730. The object may be updated and written to flash memory 740. The current original block may be dropped from the backup FIFO 750. The update process may begin processing of the next object to be updated 760.

[0094] Although a system and method according to the present invention has been described in connection with the preferred embodiment, it is not intended to be limited to the specific form set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents, as can be reasonably included within

the spirit and scope of the invention as defined by this disclosure and the appended diagrams. It is intended that the scope of the invention be limited not with this detailed description, but rather by the claims appended hereto.